PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-248941
(43)Date of publication of application : 26.09.1995

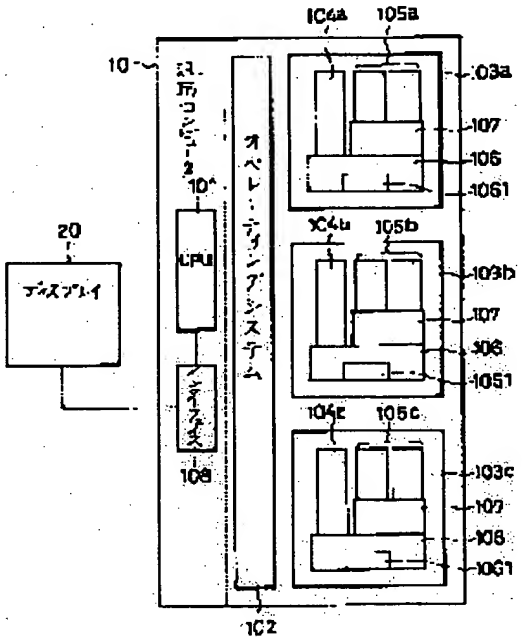
(51)Int.Cl. G06F 11/28
G06F 9/455

(21)Application number : 06-062203
(22)Date of filing : 08.03.1994

(71)Applicant : NEC CORP
(72)Inventor : MIYAMOTO TAKESHI

(54) DEBUG SUPPORT DEVICE

(57)Abstract:
PURPOSE: To perform debugging without using a target machine even in the case of software incorporated in plural target machines in a network environment by preparing the pseudo-environment of the plural target machines by using a general purpose computer.
CONSTITUTION: First, three pseudo target parts 103a, 103b and 103c defined as test objects are activated. Then, incorporated softwares 105a, 105b and 105c of the test objects are successively activated. The activated incorporated softwares 105a, 105b and 105c map respective pseudo on-line OS parts 106 and shared library parts 107 to prescribed virtual addresses, that is, 'turn them to a connection state. Thus, the incorporated softwares 105a, 105b and 105c of the respective test objects are provided with system calls provided by an on-line OS in the actual target machine and inter-processor communication function activation and shared function calls by the pseudo on-line OS parts 106 and an operation in the actual target machine is realized.



LEGAL STATUS

[Date of request for examination] 15.11.1995
[Date of sending the examiner's decision of rejection] 24.11.1998
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application] Especially this invention relates to the debugging exchange equipment which enables the trial of the inclusion software on the multiprocessor used as a test objective on a general purpose computer about the debugging exchange equipment of inclusion software.
[0002]
[Description of the Prior Art] Debugging of the inclusion software for operating it builds the software into the exchange which is usually a target machine, it is making it actually operate, detects [it includes in the exchange etc., and] malfunction etc., and it is carrying out by detecting fault (bug) of the incorporated software (debugging). Here, debugging of the software built into two or more target machines connected in the network etc. prepares two or more target machines, constructs the software to debug and is actually performed by performing this.
[0003]
[Problem(s) to be Solved by the Invention] Since it was constituted as mentioned above conventionally, in the case of the target machine with which it relates plurality for each other, and the object incorporating software operates, there was a problem of could not take test time freely but barring the increase in efficiency to the trial of inclusion software with a number limit of a target machine and the stability of a target machine. Here, there is debugging exchange equipment which enables debugging of inclusion software on a general purpose computer, without building the software for debugging into a target machine (refer to JP,1-279334,A). However, since a target machine is with this debugging exchange equipment on the assumption that debugging of unit testing of inclusion software which operates independently, about the case where there are two or more target machines by the network environment, using is impossible.
[0004] This invention is made in order to cancel the above troubles, and even if it is software built into two or more target machines which can be set to a network environment, it aims at enabling it to debug without using a target machine.
[0005]
[Means for Solving the Problem] Debugging exchange equipment of this invention maps data within a program to perform in virtual address space of arbitration first. A process control function in which execution control which is made to make accessible mapped data from all processes, and includes other R/W of memory / the contents of a register, and transit/halt of a process from a certain process can be directed, The false target section which is managed with this operating system by general purpose computer possessing an operating system which has an interprocess communication function to realize transmission and reception of data between processes, and operates on this, An inclusion software attaching part holding software for debugging with which this false target section is incorporated and used for a target machine, When it operates on an operating system of a general purpose computer and software is performed An interprocess communication function which performs in false processing to a system call which an operating system which has communication facility in a target machine which publishes a process of a ** target machine offers, and an operating system of a general purpose computer has A false online operating system which can be started, A share library

which are data which can be accessed from the software, and the aggregate of a set of functions when software is carried as an online program of a target machine, It corresponds to software which managed a process identification child who identifies an activation process by software held at an inclusion software attaching part, and had monitor directions. Processor control functions of an operating system of a general purpose computer are started. Software as a process on an operating system of a general purpose computer Make it perform, enable all processes to access the false online OS section and the share library section, perform execution control including R/W of memory and the contents of a register to all processes, and activation and a halt, and it has the monitor section which outputs debug information acquired as a result. When more than one operate the false target section on an operating system of a general purpose computer, It is characterized by communicating among the false target sections as well as actuation in a target machine, and being able to transmit and receive data with each false online operating system.

[0006]

[Function] It can communicate using a general purpose computer in the false environment of two or more target machines made in this.

[0007]

[Example] One example of this invention is explained with reference to drawing below. Drawing 1 is the block diagram showing the configuration of debugging exchange equipment in the one example of this invention. In this drawing, it is the display which is a display means [in / 10 and / in 20 / a general purpose computer 10]. [a general purpose computer]

[0008] This general purpose computer 10 has same CPU101 as that for which the actually operated target machine uses inclusion software, and operates with an operating system 102. And an operating system (OS) 102 provides a general purpose computer 10 with the following functions. First, the data within a program is mapped in the virtual address of arbitration, it is made accessible from all the processes of the mapped data, and R/W of the memory / the contents of a register to other processes [process / a certain] and the process control function in which execution control including program execution/halt can be directed are offered. And the interprocess communication function to realize transmission and reception of the data between processes is offered.

[0009] Moreover, in drawing 1 , 103a, and b and c are set on the memory which CPU101 of a general purpose computer 10 has. The false target section which ******(ed) environment of a target machine in false to the virtual space as for which OS102 is carrying out management employment, 104a, and b and c are prepared in false target section 103a, b, and c, respectively. The monitor section which performs the monitor of each actuation of false target section 103a, and b and c etc., 105a, and b and c are the inclusion software for [which is included in false target section 103a, and b and c, respectively] debugging.

[0010] using the process-control function which monitor section 104a, and b and c manage the process-identification child who identifies two or more inclusion software 105a, and b and c, and OS102 has — two or more inclusion software 105a, and b and c — the writing of memory / the contents of a register which is alike, respectively and receives, inclusion software 105a, and b and c — each activation/halt, etc. controls and the debug information which it is as a result of activation actuation outputs.

[0011] Moreover, 106 is the false online OS section which can process the system call offered by OS102 which inclusion software 105a, and b and c require. This false online OS section 106 possesses the interprocessor communication function 1061 which makes possible data communication between inclusion software 105a in different false target section 103a, b, and c, b, and c using the interprocess communication function which OS102 has.

[0012] And the share library section for [which 107 consists of the data and the sets of functions which inclusion software 105a, and b and c actually access, and is included in false target section 103a, and b and c] debugging, and 108 are interfaces which perform intermediation in signal transfer with CPU101, and a display 20 and an operator. An interface 108 inputs the debugging directions from the operator who examines, it transmits to the monitor section 104 of the false target section 103 which an operator directs, or it receives the response

from the monitor section 104 of each false target section 103, edits it, and displays it on a display 20 as debug information.
[0013] In addition, in order for the inclusion software 105 of a test objective to run on OS102, the start-up routine to drive is included in the initial starting process which is the part. The false online OS section 106 and the share library section 107 are mapped by this start-up routine in the predetermined virtual address, and reference of them is attained by it.
[0014] Next, actuation of this example is explained. Here, the case where the number of the target machines of a test objective is three is explained. First, number [which is made into a test objective], i.e., the three false target sections, 103a, and b and c are started. This is sending directions to each monitor section 104a, and b and c, and each monitor section 104a, and b and c are performed by starting each false target section 103a, and b and c using the process control function which OS102 has. And it is carried out by choosing an initial bootstrap from selected false target section 103a, each inclusion software 105a of b and c, and b and c.
[0015] In order to refer to the process information of inclusion software 105a of the test objective which the false online OS section 106 which operates on false target section 103a and b to which each monitor section 104a, and b and c correspond in that case, and c manages, and b and c, the false online OS section 106 in the predetermined virtual address is detected, and it connects with this. In other words, the false online OS section 106 is mapped by the predetermined virtual address by each monitor section 104a, and b and c.
[0016] If the initial starting process started from each monitor section 104a, and b and c is performed, in addition to the false online OS section 106, the share library section 107 is mapped by the predetermined virtual address, and the system call of false online OS section 106 offer and the supply-function call of the share library section 107 can be published from each primary stage bootstrap. and each primary stage starting process -- each false target section 103 -- sequential starting of inclusion software 105a of other test objectives in a, b, and c, and b and c is carried out.
[0017] Started inclusion software 105a, and b and c also make each false online OS section 106 and the share library section 107 mapping, i.e., a connection condition, in the predetermined virtual address. Thereby, inclusion software 105a of each test objective, and b and c will be in the condition that the system call, interprocessor communication functional starting, and the share function call which are offered with the online OS in an actual target machine are offered by the false online OS section 106, and the actuation in an actual target machine will become realizable.
[0018] As explained above, according to this example, it becomes possible to realize the test atmosphere same on a general purpose computer 10 as two or more target machines. And by this, as shown below, in a general purpose computer 10, debugging of inclusion software 105a, and b and c is attained.
[0019] First, an operator transmits debugging directions to each monitor section 104a, and b and c by directing false target section 103a made applicable to debugging, and b and c to an interface 108. By this, each monitor section 104a, and b and c collect the activation conditions of inclusion software 105a which is performing the target machine in false target section 103a and b which were completed virtually, and c, and b and c using the process control function in which OS102 possesses, and transmit to an interface 108 by making that result into debug information.
[0020] and each false target section 103 -- the interface 108 which received debug information from monitor section 104a in a, b, and c, and b and c edits the information, and displays it on a display 20. The monitor of the condition of false target section 103a by activation of inclusion software 105a of operation is carried out by monitor section 104a here, the monitor of the condition of false target section 103b by activation of inclusion software 105b of operation is carried out by monitor section 104b, and the monitor of the condition of false target section 103c by activation of inclusion software 105c of operation is carried out by monitor section 104c.
[0021] Therefore, in this debugging, since it is the bug which inclusion software 105b has probably because the fault which incorporated with false target section 103a by inclusion software 105a, and was generated during communication link actuation with false target section

103b by software 105b is the bug which inclusion software 105a has, it can distinguish and detect.

[0022] According to this example, as explained above, even if it does not use an actual target machine, using a general-purpose computer, false target machine environment is made in this, two or more activation of the inclusion software for debugging is carried out, and the monitor of these conditions can be carried out according to an individual, respectively. In addition, although considered as the same thing as that in which a target machine has CPU101 in the above-mentioned example, it does not restrict to this, and if OS102 can emulate CPU which a target machine has, what kind of thing may be used for CPU101. However, a working speed becomes slow a little in this case. Moreover, although the three false target sections were moved in the above-mentioned example, as long as the memory environment which does not restrict to this and a general purpose computer has allows, it is possible to increase the number.

[0023]

[Effect of the Invention] As explained above, even if it is the software built into two or more target machines which can be set to a network environment according to this invention, it is effective in the ability to debug without using a target machine. For this reason, since the lack of a machine time by a number limit of the target machine used as a problem and the instability of a target machine is solved at the conventional debugging production process, the effectiveness rise of a trial production process can be aimed at.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the block diagram showing the configuration of debugging exchange equipment in the one example of this invention.
[Description of Notations]
10 [-- An operating system (OS), 103a, b, c / -- The false target section, 104a, b, c / -- The monitor section, 105a, b, c / -- Inclusion software, 106 / -- The false online OS section, 107 / -- The share library section, 108 / -- Interface.] -- A general purpose computer, 20 -- A display, 101 -- CPU, 102

[Translation done.]

* NOTICES *

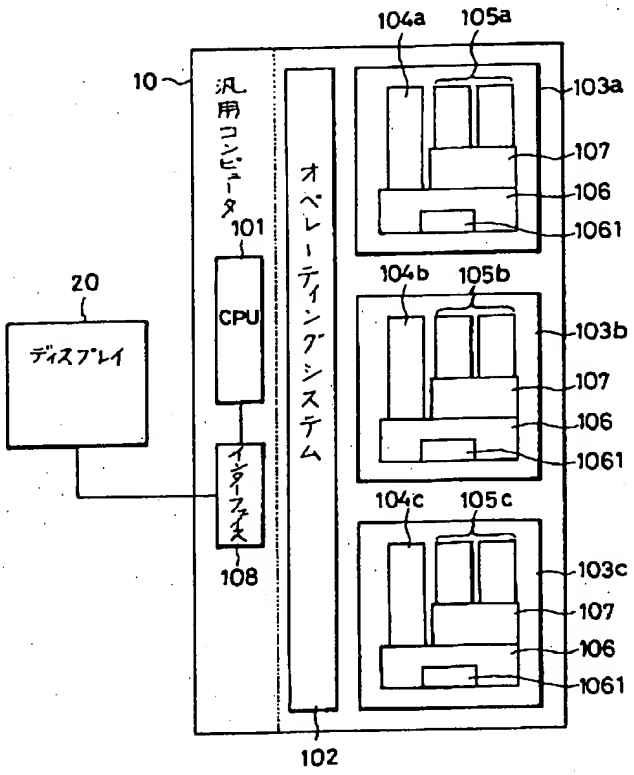
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the block diagram showing the configuration of debugging exchange equipment in the one example of this invention.
[Description of Notations]
10 [-- An operating system (OS), 103a, b, c / -- The false target section, 104a, b, c / -- The monitor section, 105a, b, c / -- Inclusion software, 106 / -- The false online OS section, 107 / -- The share library section, 108 / -- Interface.] -- A general purpose computer, 20 -- A display, 101 -- CPU, 102

[Translation done.]

Drawing selection drawing 1



[Translation done.]

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（Ａ）　(11)特許出願公開番号

特開平7－248941

(43)公開日　平成７年(1995)９月26日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　11/28	３４０　Ｃ	7313－5B		
9/455				
		7737－5B	Ｇ０６Ｆ　9/44　３１０　Ａ	

審査請求　未請求　請求項の数３　ＦＤ　（全５頁）

(21)出願番号　特願平6－62203

(22)出願日　平成６年(1994)３月８日

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号

(72)発明者　宮本　武
東京都港区芝五丁目７番１号　日本電気株式会社内

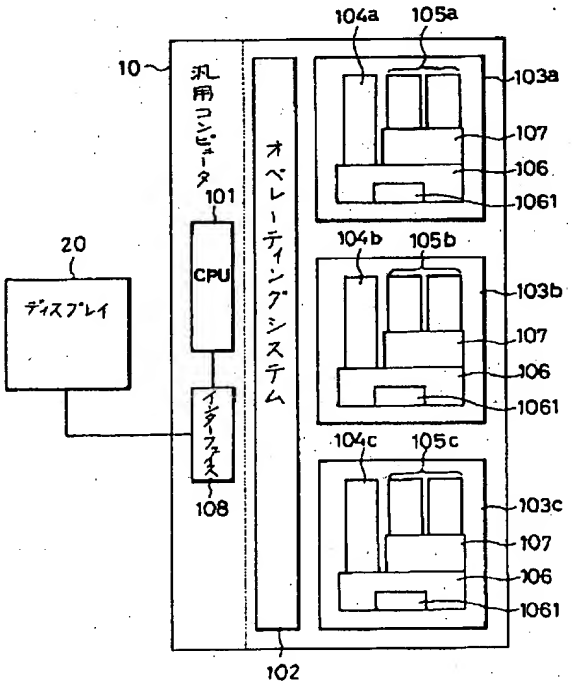
(74)代理人　弁理士　山川　政樹

(54)【発明の名称】　デバッグ支援装置

(57)【要約】

【目的】　ネットワーク環境における複数のターゲットマシンに組み込むソフトウエアであっても、ターゲットマシンを用いずにデバッグできるようにすることを目的とする。

【構成】　ＯＳ１０２が管理運用している仮想空間に、ターゲットマシンの環境を疑似的に創り出した疑似ターゲット部疑似ターゲット部１０３ａ，ｂ，ｃにおいて、各試験対象の組み込みソフトウエア１０５ａ，ｂ，ｃは、実際のターゲットマシンにおけるオンラインＯＳにより提供されるシステムコール，プロセッサ間通信機能起動および共有関数コールを、疑似オンラインＯＳ部１０６により提供される状態となる。



1

【特許請求の範囲】

【請求項１】　実行するプログラム内のデータを任意の仮想アドレス空間にマッピングし、マッピングしたデータを全てのプロセスからアクセス可能にさせ、あるプロセスから他のプロセスのメモリ／レジスタ内容の読み書きおよび走行／停止を含む実行制御を指示できるプロセス制御機能と、プロセス間のデータの送受信を実現するプロセス間通信機能を有するオペレーティングシステムを具備する汎用コンピュータに、このオペレーティングシステムにより管理されこの上で動作する疑似ターゲット部を有し、

この疑似ターゲット部が、

ターゲットマシンに組み込んで用いられるデバッグ対象のソフトウエアを保持する組み込みソフトウエア保持部と、

前記汎用コンピュータのオペレーティングシステム上で動作し、前記ソフトウエアが実行されたときの前記ターゲットマシンのプロセスが発行する前記ターゲットマシンにおける通信機能を有するオペレーティングシステムが提供するシステムコールに対する処理を疑似的に行い、前記汎用コンピュータのオペレーティングシステムが持つプロセス間通信機能を起動することができる疑似オンラインオペレーティングシステムと、

前記ソフトウエアが前記ターゲットマシンのオンラインプログラムとして搭載された際に、そのソフトウエアからアクセスできるデータおよび関数群の集合体である共有ライブラリと、

前記組み込みソフトウエア保持部に保持されたソフトウエアによる実行プロセスを識別するプロセス識別子を管理し、監視指示のあったソフトウエアに対応して、前記汎用コンピュータのオペレーティングシステムのプロセッサ制御機能を起動して、前記ソフトウエアを前記汎用コンピュータのオペレーティングシステム上のプロセスとして実行させ、前記疑似オンラインオペレーティングシステムと共有ライブラリ部に全てのプロセスがアクセスすることを可能にさせ、前記すべてのプロセスに対するメモリ，レジスタ内容の読み書きと実行，停止とを含む実行制御を行い、その結果得られるデバッグ情報を出力するモニタ部とを有し、

前記汎用コンピュータのオペレーティングシステム上で前記疑似ターゲット部を複数動作したとき、それぞれの前記疑似オンラインオペレーティングシステムにより、前記ターゲットマシンにおける動作と同様に、前記疑似ターゲット部同士の間で通信を行い、データの送受信が行えることを特徴とするデバッグ支援装置。

【請求項２】　請求項１記載のデバッグ支援装置において、

前記汎用コンピュータのオペレーティングシステムが、前記ターゲットマシンのＣＰＵの動作をエミュレートできることを特徴とするデバッグ支援装置。

2

【請求項３】　請求項１記載のデバッグ支援装置において、

前記汎用コンピュータが、前記ターゲットマシンと実質的に同一なＣＰＵを備えていることを特徴とするデバッグ支援装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】この発明は、組み込みソフトウエアのデバッグ支援装置に関し、特に試験対象となるマルチプロセッサ上の組み込みソフトウエアの試験を汎用コンピュータ上で可能にさせるデバッグ支援装置に関する。

【０００２】

【従来の技術】交換機などに組み込み、それを動作させるための組み込みソフトウエアのデバッグは、通常ターゲットマシンである交換機にそのソフトウエアを組み込んで、実際に動作させることで、誤動作などを検出して、組み込んだソフトウエアの不具合（バグ）を検出（デバッグ）することで行っている。ここで、ネットワークなどで接続されている複数のターゲットマシンに組み込むソフトウエアのデバッグは、実際に複数のターゲットマシンを用意して、デバッグするソフトウエアを組み、これを実行させることで行う。

【０００３】

【発明が解決しようとする課題】従来は以上のように構成されていたので、ソフトウエアを組み込む対象が、複数が関連し合って動作するターゲットマシンの場合、ターゲットマシンの数制限およびターゲットマシンの安定度により、自由に試験時間をとれず、組み込みソフトウエアの試験に対する効率化を妨げるという問題があった。ここで、ターゲットマシンにデバッグ対象のソフトウエアを組み込むこと無く、汎用コンピュータ上で、組み込みソフトウエアのデバッグを可能にするデバッグ支援装置がある（特開平１－２７９３３４号公報参照）。しかし、このデバッグ支援装置では、ターゲットマシンが単独で動作する組み込みソフトウエアの単体試験のデバッグを前提にしいるので、ネットワーク環境でターゲットマシンが複数ある場合については、用いることが不可能である。

【０００４】この発明は、以上のような問題点を解消するためになされたものであり、ネットワーク環境における複数のターゲットマシンに組み込むソフトウエアであっても、ターゲットマシンを用いずにデバッグできるようにすることを目的とする。

【０００５】

【課題を解決するための手段】この発明のデバッグ支援装置は、まず、実行するプログラム内のデータを任意の仮想アドレス空間にマッピングし、マッピングしたデータを全てのプロセスからアクセス可能にさせ、あるプロセスから他のプロセスのメモリ／レジスタ内容の読み書

きおよび走行／停止を含む実行制御を指示できるプロセス制御機能と、プロセス間のデータの送受信を実現するプロセス間通信機能を有するオペレーティングシステムを具備する汎用コンピュータにこのオペレーティングシステムにより管理されこの上で動作する疑似ターゲット部と、この疑似ターゲット部がターゲットマシンに組み込んで用いられるデバッグ対象のソフトウエアを保持する組み込みソフトウエア保持部と、汎用コンピュータのオペレーティングシステム上で動作してソフトウエアが実行されたときのターゲットマシンのプロセスが発行するターゲットマシンにおける通信機能を有するオペレーティングシステムが提供するシステムコールに対する処理を疑似的に行って汎用コンピュータのオペレーティングシステムが持つプロセス間通信機能を起動することができる疑似オンラインオペレーティングシステムと、ソフトウエアがターゲットマシンのオンラインプログラムとして搭載された際にそのソフトウエアからアクセスできるデータおよび関数群の集合体である共有ライブラリと、組み込みソフトウエア保持部に保持されたソフトウエアによる実行プロセスを識別するプロセス識別子を管理し，監視指示のあったソフトウエアに対応して汎用コンピュータのオペレーティングシステムのプロセッサ制御機能を起動し，ソフトウエアを汎用コンピュータのオペレーティングシステム上のプロセスとして実行させ，疑似オンラインＯＳ部と共有ライブラリ部に全てのプロセスがアクセスすることを可能にさせ，すべてのプロセスに対するメモリ，レジスタ内容の読み書きと実行，停止とを含む実行制御を行い，その結果得られるデバッグ情報を出力するモニタ部とを有し、汎用コンピュータのオペレーティングシステム上で疑似ターゲット部を複数動作したとき、それぞれの疑似オンラインオペレーティングシステムにより、ターゲットマシンにおける動作と同様に、疑似ターゲット部同士の間で通信を行い、データの送受信が行えることを特徴とする。

【０００６】

【作用】汎用コンピュータを用いて、この中に作り出された複数のターゲットマシンの疑似環境同士で通信できる。

【０００７】

【実施例】以下この発明の１実施例を図を参照して説明する。図１は、この発明の１実施例でデバッグ支援装置の構成を示す構成図である。同図において、１０は汎用コンピュータ、２０は汎用コンピュータ１０における表示手段であるディスプレイである。

【０００８】この汎用コンピュータ１０は組み込みソフトウエアを実際に動作させるターゲットマシンが用いるものと同一のＣＰＵ１０１を有し、オペレーティングシステム１０２により動作する。そして、オペレーティングシステム（ＯＳ）１０２は、汎用コンピュータ１０に以下の機能を提供するものである。まず、プログラム内のデータを任意の仮想アドレスにマッピングし、マッピングしたデータの全てのプロセスからアクセス可能にさせ、あるプロセスから他のプロセスへのメモリ／レジスタ内容の読み書きや、プログラムの実行／停止を含む実行制御を指示できるプロセス制御機能を提供する。そして、プロセス間のデータの送受信を実現するプロセス間通信機能を提供するものである。

【０００９】また、図１において、１０３ａ，ｂ，ｃは汎用コンピュータ１０のＣＰＵ１０１が有するメモリ上において、ＯＳ１０２が管理運用している仮想空間に、ターゲットマシンの環境を疑似的に創り出した疑似ターゲット部、１０４ａ，ｂ，ｃは疑似ターゲット部１０３ａ，ｂ，ｃ内にそれぞれ設けられ、疑似ターゲット部１０３ａ、ｂ、ｃのそれぞれの動作の監視などを行うモニタ部、１０５ａ，ｂ，ｃは疑似ターゲット部１０３ａ，ｂ，ｃにそれぞれ組み込まれるデバッグ対象の組み込みソフトウエアである。

【００１０】モニタ部１０４ａ，ｂ，ｃは、複数の組み込みソフトウエア１０５ａ，ｂ，ｃを識別するプロセス識別子を管理し、ＯＳ１０２が有しているプロセス制御機能を使用することにより、複数の組み込みソフトウエア１０５ａ，ｂ，ｃそれぞれに対するメモリ／レジスタ内容の書き込みや、組み込みソフトウエア１０５ａ，ｂ，ｃそれぞれの実行／停止などの制御を行い、実行動作結果であるデバッグ情報を出力する。

【００１１】また、１０６は組み込みソフトウエア１０５ａ，ｂ，ｃが要求するＯＳ１０２によって提供されるシステムコールを処理することができる疑似オンラインＯＳ部である。この疑似オンラインＯＳ部１０６は、ＯＳ１０２が持つプロセス間通信機能を利用して、異なる疑似ターゲット部１０３ａ、ｂ、ｃ内の組み込みソフトウエア１０５ａ，ｂ，ｃ間のデータ通信を可能とするプロセッサ間通信機能１０６１を具備する。

【００１２】そして、１０７は組み込みソフトウエア１０５ａ，ｂ，ｃが実際にアクセスするデータおよび関数群から構成され、疑似ターゲット部１０３ａ，ｂ，ｃに組み込まれるデバッグ対象の共有ライブラリ部、１０８はＣＰＵ１０１とディスプレイ２０や操作者との信号授受における仲立ちを行うインターフェイスである。インターフェイス１０８は試験を行う操作者からのデバッグ指示を入力し、操作者が指示する疑似ターゲット部１０３のモニタ部１０４に送信したり、各疑似ターゲット部１０３のモニタ部１０４からの応答を受信し、それを編集し、ディスプレイ２０にデバッグ情報として表示する。

【００１３】なお、試験対象の組み込みソフトウエア１０５は、ＯＳ１０２上で走行するために、その一部である初期起動プロセスに、駆動されるスタートアップルーチンが組み込まれている。このスタートアップルーチンにより、疑似オンラインＯＳ部１０６および共有ライブ

ラリ部１０７が所定の仮想アドレスにマッピングされ参照可能となる。

【００１４】次に、本実施例の動作について説明する。ここでは、試験対象のターゲットマシンが３つの場合について説明する。まず、試験対象とする数だけ、すなわち３つの疑似ターゲット部１０３ａ，ｂ，ｃを起動させる。これは、各モニタ部１０４ａ，ｂ，ｃに対して指示を送ることで、各モニタ部１０４ａ，ｂ，ｃがＯＳ１０２が有するプロセス制御機能を使って、それぞれの疑似ターゲット部１０３ａ，ｂ，ｃを起動させることで行われる。そして、選択した疑似ターゲット部１０３ａ，ｂ，ｃのそれぞれの組み込みソフトウエア１０５ａ，ｂ，ｃの中から初期起動プログラムを選択することで行われる。

【００１５】その際、各モニタ部１０４ａ，ｂ，ｃは、対応する疑似ターゲット部１０３ａ，ｂ，ｃ上で動作する疑似オンラインＯＳ部１０６が管理する試験対象の組み込みソフトウエア１０５ａ，ｂ，ｃのプロセス情報を参照するために、所定の仮想アドレスにある疑似オンラインＯＳ部１０６を検出し、これに接続する。言い替えると、疑似オンラインＯＳ部１０６が各モニタ部１０４ａ，ｂ，ｃにより所定の仮想アドレスにマッピングされる。

【００１６】各モニタ部１０４ａ，ｂ，ｃから起動された初期起動プロセスが実行されると、疑似オンラインＯＳ部１０６に加え共有ライブラリ部１０７が所定の仮想アドレスにマッピングされ、各初期起動プログラムから疑似オンラインＯＳ部１０６提供のシステムコールおよび共有ライブラリ部１０７の供給関数コールが発行できるようになる。そして、各初期起動プロセスは、各疑似ターゲット部１０３ａ，ｂ，ｃ内の他の試験対象の組み込みソフトウエア１０５ａ，ｂ，ｃを順次起動する。

【００１７】起動された組み込みソフトウエア１０５ａ，ｂ，ｃも、それぞれの疑似オンラインＯＳ部１０６および共有ライブラリ部１０７を所定の仮想アドレスにマッピング、すなわち、接続状態とする。これにより、各試験対象の組み込みソフトウエア１０５ａ，ｂ，ｃは、実際のターゲットマシンにおけるオンラインＯＳにより提供されるシステムコール，プロセッサ間通信機能起動および共有関数コールを、疑似オンラインＯＳ部１０６により提供される状態となり、実際のターゲットマシンにおける動作が実現可能になる。

【００１８】以上説明したように、本実施例によれば、汎用コンピュータ１０上に複数ターゲットマシンと同じ試験環境を実現することが可能となる。そして、このことにより、以下に示すように、汎用コンピュータ１０において、組み込みソフトウエア１０５ａ，ｂ，ｃのデバッグが可能となる。

【００１９】まず、操作者がインターフェイス１０８にデバッグ対象とする疑似ターゲット部１０３ａ，ｂ，ｃを指示することで、それぞれのモニタ部１０４ａ，ｂ，ｃにデバッグ指示を送信する。このことにより、各モニタ部１０４ａ，ｂ，ｃは、ターゲットマシンを仮想的に作り上げた疑似ターゲット部１０３ａ，ｂ，ｃ内において、実行している組み込みソフトウエア１０５ａ，ｂ，ｃの実行状況を、ＯＳ１０２が具備するプロセス制御機能を使って収集し、その結果をデバッグ情報としてインターフェイス１０８に送信する。

【００２０】そして、各疑似ターゲット部１０３ａ，ｂ，ｃ内のモニタ部１０４ａ，ｂ，ｃからデバッグ情報を受信したインターフェイス１０８は、その情報を編集してディスプレイ２０に表示する。ここで、組み込みソフトウエア１０５ａの実行による疑似ターゲット部１０３ａの動作状況は、モニタ部１０４ａによってモニタされ、組み込みソフトウエア１０５ｂの実行による疑似ターゲット部１０３ｂの動作状況は、モニタ部１０４ｂによってモニタされ、組み込みソフトウエア１０５ｃの実行による疑似ターゲット部１０３ｃの動作状況は、モニタ部１０４ｃによってモニタされている。

【００２１】従って、このデバッグにおいては、組み込みソフトウエア１０５ａによる疑似ターゲット部１０３ａと組み込みソフトウエア１０５ｂによる疑似ターゲット部１０３ｂとの通信動作中に発生した不具合が、組み込みソフトウエア１０５ａが有するバグのためか、組み込みソフトウエア１０５ｂが有するバグのためかが、区別して検出できるものである。

【００２２】以上説明したように、この実施例によれば、実際のターゲットマシンを用いなくても、汎用的なコンピュータを用いて、この中に疑似的なターゲットマシン環境を作り出し、デバッグ対象の組み込みソフトウエアを複数実行させ、これらの状況をそれぞれ個別にモニタできる。なお、上記実施例では、ＣＰＵ１０１をターゲットマシンが有するものと同一のものとしたが、これに限るものではなく、ＯＳ１０２がターゲットマシンの有するＣＰＵをエミュレートできるものなら、ＣＰＵ１０１にどのようなものを用いても良い。ただし、この場合は、動作速度が若干遅くなる。また、上記実施例では、疑似ターゲット部を３つ動かすようにしたが、これに限るものではなく、汎用コンピュータの有するメモリ環境が許す限り、その数を増やすことが可能である。

【００２３】

【発明の効果】以上説明したように、この発明によれば、ネットワーク環境における複数のターゲットマシンに組み込むソフトウエアであっても、ターゲットマシンを用いずにデバッグできるという効果がある。このため、従来のデバッグ工程で、問題となっていたターゲットマシンの数制限、および、ターゲットマシンの不安定によるマシン使用時間不足が解決されるため、試験工程の効率アップが図れる。

【図面の簡単な説明】

【図１】　この発明の１実施例でデバッグ支援装置の構成を示す構成図である。

【符号の説明】

１０…汎用コンピュータ、２０…ディスプレイ、１０１…ＣＰＵ、１０２…オペレーティングシステム（ＯＳ）、１０３ａ，ｂ，ｃ…疑似ターゲット部、１０４ａ，ｂ，ｃ…モニタ部、１０５ａ，ｂ，ｃ…組み込みソフトウエア、１０６…疑似オンラインＯＳ部、１０７…共有ライブラリ部、１０８…インターフェイス。

【図１】